Open Source IoT 보안을 위한 익스플로잇 취약점 분석 모듈 설계

박근일*, 박상현*, 전문석*
*숭실대학교 컴퓨터학과
e-mail:higa_ps15@ssu.ac.kr
shyeon15@ssu.ac.kr
mjun@ssu.ac.kr

A Design of Exploit Vulnerability Analysis Module for Open Source IoT Security

Geunil Park*, Sanghyeon Park*, Moon-Seog Jun*
*Dept of Computer Science & Engineering, Soongsil University

요 약

최근 컴퓨터, 휴대폰 등 전자기기만 인터넷 연결이 가능하던 시대를 지나 냉장고, 에어컨, 현관문 등 모든 종류의 사물들 간 사람의 개입이 필요 없는 초연결사회로 발전하고 있다. 이러한 모든 사물들 간 인터넷기반으로 상호 연결되어있는 IoT(Internet of Things)환경이 급격히 성장 하고 있는 가운데 더불어 OSIoT(Open Source IoT)의 수요도 함께 급성장하고 있다. OSIoT의 소프트웨어는 보안에 대한 전문적인 개발자의 체계적인 설계에 의해 개발되어야만 하트블리드(HeartBleed), 쉘쇼크(ShellShock)와 같은 다양한 보안취약점에 안전하다. 하지만 OSIoT소프트웨어는 누구나 쉽게 접근 설계가 가능하기 때문에 일반적으로 배포되고 있는 OSIoT의 소프트웨어 검증이 필요하다. 따라서 본 논문에서는 다른 소프트웨어 점검 도구들과 연계 가능한 정적분석 도구인 취약점 별 익스플로잇 적용 모듈 설계를 제안한다.

1. 서론

최근 인터넷을 기반으로 사람과 사람, 사람과 사물 등 생활 속의 모든 것들을 상호연결 시키는 IoT(Internet of Things)기술은 2020년이면 500억개 이상의 사물들이 다양한 정보를 공유하는 초연결사회의 스마트환경이 주목되는 가운데, 소프트웨어의 제작자의 권리를 지키면서 원시코드를 누구나 연람할 수 있도록 한 OSIoT(Open Source IoT)을 사용하여 새로운 시장을 창출하고 있다. OSIoT의 소프트웨어의 소스코드는 쉽고 누구나 접근, 설계가 가능하지만 보안 취약점으로부터 안전한 소프트웨어위해 체계적인 설계 가능한 전문가에 의해 개발되어야한다. 하지만 보통 전문지식이 많지 않은 개발자에 의해 손쉽게 만들어지고 배포되기 때문에, 체계적인 설계가 아닌 OSIoT는 하트블러드 또는 쉘쇼크 등과 같은 보안 취약점이 발생하고 이를 통해 다양한 공격으로 이어진다. 이에 본 논문에서는 보안 취약점에 안전한 OSIoT를 제공하기 위하여 OSIoT의 소스코드 정적분석도구인 취약점 별 익스플로잇 분석 모듈을 설계를 제안한다.

2. 오픈소스 사물인터넷(OSIoT)

OSIoT의 기본적인 목적은 수많은 사물인터넷 구성 요소들의 갖고 있는 이질성을 극복하고, 사물인터넷의 핵심인 기기 간 연결성과 상호 호환성을 확보하여 동작할 수 있도록 함으로써, 보다 유용한 사물인터넷 응용과 서비스를 제공할 수 있도록 하는데 있다. 오픈소스 사물인터넷 환경을 통해 기대할 수 있는 효과들은 다음과 같다.

낮은 진입비용 : 일반적으로 오픈소스는 대부분 별도의 비용 없이 다운로드 및 소스코드 수정·재배포가 가능한 것이 특징이다. 따라서 초기 개발 비용이 적게 요구 된다는 장점이 있다.

빠르고 유연한 개발 : 오픈소스 프로젝트는 보통 최신 기술정보 및 문제점과 해결책을 공유하는 형태로 자유롭게 운영되기에 독점 프로그램에 비해 기술 혁신 속도가 빠르다.

결합 확장성 : 오픈소스 프로젝트들은 다른 오픈소스 프로젝트들을 함께 활용함으로써 새로운 응용들을 손쉽게 확장 개발 가능할 수 있다. 또한 다양한 정보를 개발자들에게 손쉽게 제공할 수 있어 다양한 응용 개발이 빠르게 진행될 수 있다.

다양한 기술 기여도: 오픈소스 프로젝트에는 다양한 멤버들이 참여하여 자유롭게 자신의 생각과 요구사항에 맞는 제안과 기여를 하기에 다양한 관점을 수용하는 변화가 가능하다.

오픈 포맷과 프로토콜: 오픈소스 프로젝트는 주로 오픈포

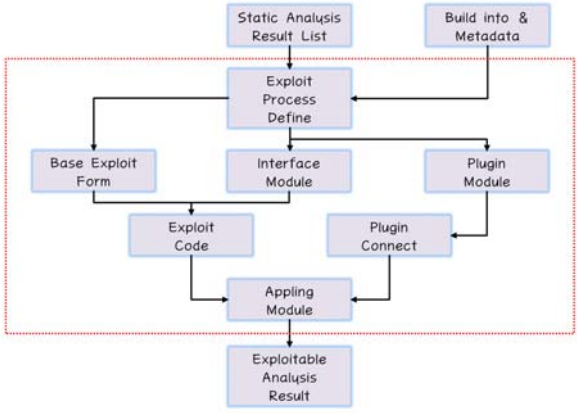
This work was supported by the ICT R&D program of MSIP/IITP. [R0112-14-1061, The analysis technology of a vulnerability on an open-source software, and the development of Platform]

맷 또는 프로토콜을 사용하기 때문에 서로 다른 SW간 상호 연동성이 보장되는 장점이 있다. 다수의 기기들이 서로 다른 네트워크를 통해 연결되는 사물인터넷 환경은 오픈소스 기반으로 보다 유연하게 연동될 수 있다.

신뢰성과 안정성: 오픈소스 개발 과정에는 전 세계에 있는 수많은 우수한 개발자들이직접개발과 디버깅 과정에 참여하기 때문에 In-house에서 폐쇄적으로 개발되는 독점 프로그램에 비해 비교적 안정적으로 동작 가능하다. 그러나 이러한 신뢰성과 안정성은 많은 개발자들의 적극적인 참여가 있을 때에만 가능한 것이다.

3. 제안

제안하는 OSIoT를 오픈 소스코드에 발생하는 취약점으로부터 안전하게 보안하기 위해 정적분석의 취약점 별 익스플로잇 분석 모듈을 설계한다. (그림 1)과 같이 익스플로잇터블이 검증된 취약점 리스트들을 분석하여 각 취약점에 대한 익스플로잇을 생성하기 위한 절차 설계하고 동작 환경을 특정하여 어플리케이션의 빌드정보와 어플리케이션을 실행한 후 취약점 코드 부분까지 도달할 수 있는 정보를 제공한다.

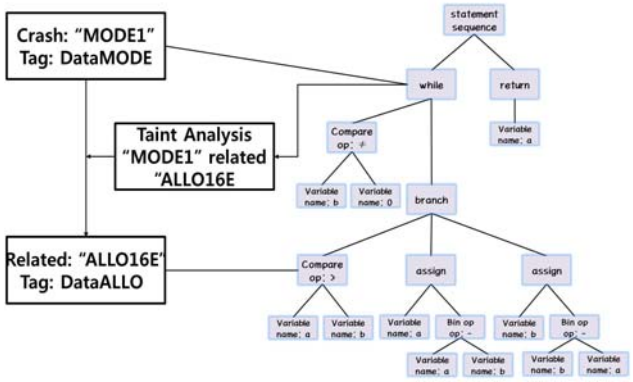


(그림 1) 익스플로잇 생성을 위한 절차

정적 모듈의 취약한 소스코드, 공격의 취약점, 소프트웨어구조 등 전송을 하고 취약점과 소스코드에서 페이로드와 기능 위치(소스코드-런타임 분석)를 도출하여 적용 가능한 익스플로잇 판단한다. 익스플로잇의 기본 수행 구조, 런타임 적용을 위한 인터페이스 모듈을 생성하고, 페이로드 리스트를 적용하도록 설정과 플러그인 등 추가 구성요소를 확인하여 설정하게 되면 취약점에 대한 공격을 수행하여 결과 도출 및 분석 할 수 있다.

4. 안전성평가

(그림 2)는 분석된 기능별 데이터 흐름 상 앞쪽에서 발생한 DataMODE 데이터 형식의 데이터 “MODE1”의 충돌로 연관 데이터인 “ALLO16E”의 fuzzing Test data suite인 DataALLO를 도출하여 기능 흐름상 관련이 있는 후속 기능에서 read access violation과 관련된 Test data suite를 사용한다는 것을 검출해내고, 데이터“MODE1”의 읽기위반 오류로 인하여 영향을 받는 데이터“ALLO16E”가 다른 기능에서 활용되는 취약점 시나리오이다. 익스플로잇을 취약점을 일으키는 페이로드를 적용하는 어플리케이션으로 정의하고, 익스플로잇터블 검증 단계에서 검출된 어플리케이션의 빌드 및 동작 정보와 취약점정보에 따라 분석 어플리케이션을 대상으로 실행 가능한 실행 어플리케이션 구성하여 분석한다.



(그림 2) 소프트웨어 기능 흐름에 따른 익스플로잇 예시

5. 결론

IoT는 다양한 사물이 사람의 개입이 없이 서로 통신되는 사용자 중심의 거대한 환경이라 할 수 있다. 수많은 형태의 프로토콜과 플랫폼, 소프트웨어와 하드웨어 기술들이 결합해야하는 환경은 결코 단일 기업의 폐쇄 적인 기술로 개발할 수 없는 것이며, 개방형 협력 모델을 기반으로 하는 OSIoT이 절대적으로 필요 하지만 OSIoT는 누구에게나 소스코드가 공개되어있기 때문에 소프트웨어의 보안취약점을 이용한 보안사고의 위험도 커지고 있다. 본 논문에서 제안한 OSIoT의 소스코드를 체계적인 보안하기 위해 취약점 별 익스플로잇 분석 모듈 설계하여 검증함으로써 취약점 악용 시나리오를 분석할 수 있다. 향후 본 논문을 기반으로 다른 세부 요소를 보완하고 추가함으로써 OSIoT에 대한 취약점 관리를 위한 연구를 기대한다.

참고문헌

[1] 최성찬, et al. "사물인터넷 플랫폼 오픈소스 동향." 한국통신학회지 (정보와통신) 32.5 (2015): 16-22.

[2] 김형주, et al. "융복합 전자정부 서비스를 위한 전자정부 표준프레임워크 기반 시큐어코딩 점검 시스템 설계 및 개발." 디지털융복합연구 13.3 (2015): 201-208.

[3] Andreasen, Morten Sieker, et al. "Usability in open source software development: Opinions and practice." Information technology and control 35.3 (2015).

[4] Barnum, S. "Common attack pattern enumeration and classification (capec) schema description." Cigital Inc, http://capec. mitre.org/documents/documentation/CAPEC_Schema_Descr iption_v1 3 (2008).